

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

BYEONG-CHUL KIM

Serial No.: *to be assigned*

Examiner: *to be assigned*

Filed: 17 September 2003

Art Unit: *to be assigned*

For: SYSTEM COMBINED WITH LOADSHARING STRUCTURE AND
PRIMARY/BACKUP STRUCTURE

CLAIM OF PRIORITY UNDER 35 U.S.C. §119

Mail Stop : Application Number

Commissioner for Patents

P.O. Box 1450

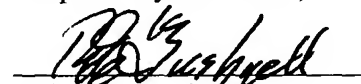
Alexandria, VA 22313-1450

Sir:

The benefit of the filing date of the following prior foreign application, Korean Priority No.2002-62842 (filed in Korea on 15 October 2002) and filed in the U.S. Patent and Trademark Office on 17 September 2003 is hereby requested and the right of priority provided in 35 U.S.C. §119 is hereby claimed.

In support of this claim, filed herewith is certified copies of said original foreign applications.

Respectfully submitted,



Robert E. Bushnell

Reg. No.: 27,774

Attorney for the Applicant

1522 "K" Street, N.W., Suite 300
Washington, D.C. 20005
(202) 408-9040

Folio: P56919

Date: 9/17/03

I.D.: REB/rfc

대한민국 특허청
KOREAN INTELLECTUAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2002-0062842
Application Number

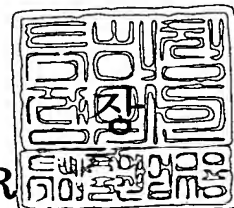
출원년월일 : 2002년 10월 15일
Date of Application OCT 15, 2002

출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 02 월 27 일

특 허 청
COMMISSIONER



【서지사항】

【서류명】 특허출원서
【권리구분】 특허
【수신처】 특허청장
【제출일자】 2002.10.15
【발명의 명칭】 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템
【발명의 영문명칭】 System combined loadsharing structure and primary/backup structure
【출원인】
【명칭】 삼성전자 주식회사
【출원인코드】 1-1998-104271-3
【대리인】
【성명】 박상수
【대리인코드】 9-1998-000642-5
【포괄위임등록번호】 2000-054081-9
【발명자】
【성명의 국문표기】 김병철
【성명의 영문표기】 KIM,BYEONG CHEOL
【주민등록번호】 680706-1540322
【우편번호】 442-470
【주소】 경기도 수원시 팔달구 영통동 967-2 풍림아파트 603동 40호
【국적】 KR
【심사청구】 청구
【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 박상수 (인)
【수수료】
【기본출원료】 17 면 29,000 원
【가산출원료】 0 면 0 원
【우선권주장료】 0 건 0 원
【심사청구료】 5 항 269,000 원
【합계】 298,000 원
【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템에 관한 것으로서, 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템은 각각의 시스템마다 프라이머리부와 백업부를 두고; 이벤트 처리는 모든 시스템의 프라이머리에서 기본적으로 수행하면서 시스템이 비정상적인 상태를 대비하여 현재 시스템이 아닌 다른 시스템에 현재 프라이머리부에서 처리하는 이벤트 처리 데이터를 복사(백업)시키고; 시스템이 다운되는 경우가 발생되면 미리 백업 시켜놓은 시스템의 백업부를 프라이머리부로 변경하여 변경된 프라이머리부에서 이전의 프라이머리부에서 수행하던 이벤트를 계속 진행하고, 필요시 점유한 자원을 복구시키며 처리해야 할 이벤트가 더 이상 존재하지 않게 되면 변경된 프라이머리부의 동작을 중지하도록 이루어져, 이벤트 발생에 대하여 부하 분담하여 처리하므로 자원의 낭비없이 성능이 향상되는 구조이며, 백업 시스템으로 데이터가 복사되므로 시스템을 안정적이며 효율적으로 운영할 수 있는 효과가 있다.

【대표도】

도 2a

【색인어】

부하분담, 백업, 이중화, 프라이머리, 분산제어

【명세서】**【발명의 명칭】**

부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템{System combined loadsharing structure and primary/backup structure}

【도면의 간단한 설명】

도 1a는 종래기술에 따라 서로 다른 시스템 사이의 프라이머리/백업 구조를 간략히 도시한 개념도,

도 1b는 종래기술에 따라 서로 다른 시스템 사이의 부하분담 구조를 간략히 도시한 개념도,

도 2a는 본 발명의 일실시예에 따른 부하분담 구조와 프라이머리/백업 구조가 결합된 형태를 간략히 도시한 개념도,

도 2b는 도 2a와 같은 형태로 이루어진 시스템에서 이벤트 발생시 부하분담 및 백업 설정 관계를 도시한 블록도,

도 3a는 도 2b와 같은 관계에 따라 실제로 특정 프라이머리가 다운된 경우에 특정 백업부가 특정 프라이머리부로 변경되는 과정을 도시한 블록도,

도 3b는 도 2b와 같은 관계에 따라 실제로 특정 프라이머리가 다운된 경우에 남은 프라이머리의 부하분담 과정을 도시한 블록도.

<도면의 주요 부분에 대한 부호 설명>

100, 200, 300 : 시스템 A, 시스템 B, 시스템 C

110, 210, 310 : 프라이머리 1, 프라이머리 2, 프라이머리 3

120, 220, 320 : 백업 3, 백업 1, 백업 2

400 : 분산 제어 환경

500 : CM(Configuration Management)

600 : 분산 알고리즘

700 : Logical Shared Resource

800 : 이벤트

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<16> 본 발명은 분산제어 방식에 관한 것으로서, 더욱 상세하게는 부하분담 구조와 이중화(프라이머리/백업:Primary/Backup)구조를 혼합시켜 시스템을 효율적이면서도 안정적으로 운영할 수 있도록 하는 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템에 관한 것이다.

<17> 일반적으로 여러 개의 물리적으로 분산된 시스템을 분산제어하는 방식은 크게 이중화 방식(이하 '프라이머리/백업'이라 함)과 부하분담 방식으로 나뉘어진다.

<18> 도 1a는 종래기술에 따라 서로 다른 시스템 사이의 프라이머리/백업 구조를 간략히 도시한 개념도이고, 도 1b는 종래기술에 따라 서로 다른 시스템 사이의 부하분담 구조를 간략히 도시한 개념도이다.

- <19> 먼저 도 1a를 참조하여 프라이머리/백업 구조에 대해 설명하면 다음과 같다.
- <20> 도 1a에 도시된 바와 같이, 프라이머리/백업 구조는 물리적으로 분산된 시스템 A, 시스템 B(10,20)에서 시스템 A(10)의 프라이머리 1(11)에서 이벤트(event) 처리를 수행하고, 시스템 A(10)의 프라이머리 1(11)이 다운되는 경우를 대비하여 진행중인 이벤트 처리 및 공유된 자원들을 복구할 때 필요한 데이터를 시스템 B(20)의 백업 1(21)로 복사하는 구조로 이루어진다.
- <21> 따라서 프라이머리 1(11)과 백업 1(21)은 데이터가 일치된다. 이렇게 데이터가 일치된 상태에서 프라이머리 1(11)이 다운되는 경우 백업 1(21)이 새로운 프라이머리가 되어 프라이머리 1(11)이 처리하던 이벤트들을 처리하므로 진행중인 이벤트 처리 및 공유된 자원들에 대해 정상적으로 동작이 가능해진다.
- <22> 그러나 상기 프라이머리/백업 구조는 발생한 이벤트 양에 상관없이 항상 프라이머리 1(11)에서 모든 이벤트를 처리하고, 백업 1(21)로만 이루어진 시스템 B(20)는 무조건 프라이머리 1(11)에서 수행하는 이벤트 처리 내용에 대한 데이터를 복사하여 대기하는 상태이므로 프라이머리 1(11)만 있는 시스템 A(10)에 부하가 집중되며, 나머지 시스템 B(20)는 활용되지 않은 상태로 낭비가 되는 문제점이 있다. 또한 프라이머리 1(11)에만 부하가 집중되므로 시스템의 성능이 저하되는 문제점이 있다.
- <23> 이하 도 1b를 참조하여 부하분담 구조를 설명하면, 부하분담 방식은 처리해야 할 이벤트들에 대해 특정한 시스템에 부하가 집중되지 않도록 부하를 균일하게 분산하여 처리하는 구조로서, 모든 시스템 즉 시스템 A,B,C(10,20,30)가 프라이머리 1, 2, 3(11,21,31)으로 동작하게 된다.

- <24> 모든 시스템(10,20,30)들이 프라이머리(11,21,31)로 동작하므로 시스템이 비정상적인 경우 발생 시, 해당 시스템이 처리하던 이벤트 및 공유된 자원들에 대해 백업 시스템이 없으므로 진행중인 이벤트 처리 및 공유된 자원들에 대해 서비스가 불가능하고 자원을 복구할 수 없는 문제점이 있다.
- <25> 상기의 문제점을 정리해보면 다음과 같다.
- <26> 먼저 프라이머리/백업 구조는 부하분담이 효율적으로 제공되는 구조가 아니며, 가용(可用)한 시스템의 자원을 모두 활용하지 않으므로 자원이 낭비 및 시스템의 효율성이 떨어지게 된다. 또한 시스템의 확장을 위해서는 프라이머리/백업과 같이 항상 한 쌍으로 증설되므로 비용을 증가시키는 문제점이 있다.
- <27> 그리고 부하분담 구조는 현재 발생하는 이벤트 양을 부하 분담 로직에 의해 균일하게 분산처리되어 시스템의 자원을 효율성있게 사용하는 긍정적인 측면이 있다. 이것은 모든 시스템이 정상적으로 동작할 경우에 해당하는 측면이고, 비정상적으로 동작하는 시스템이 발생하는 경우에는 즉 프라이머리로 동작하는 모든 시스템들 중 특정 시스템이 다운되는 경우 다운된 시스템에서 처리하던 이벤트 들에 대한 지속성을 제공할 수 없고 점유되어 사용하고 있던 공용 데이터들에 대해 불복구 현상이 발생하게 되어 시스템이 신뢰성에 문제가 된다.

【발명이 이루고자 하는 기술적 과제】

- <28> 따라서 본 발명은 상기와 같은 문제점을 해결하기 안출된 것으로서, 프라이머리/백업 구조와 부하분담 구조가 안고 있는 구조적인 문제점을 해결할 수 있도록 각각의 시스

템마다 프라이머리부와 백업부를 두고, 이벤트 처리는 모든 시스템의 프라이머리에서 기본적으로 수행하다가, 시스템이 비정상적인 상태가 되는 경우에 한하여 다른 시스템의 백업부에 현재 프라이머리부에서 처리하던 이벤트 처리 데이터를 복사시키도록 이루어진 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템을 제공하는데 그 목적이 있다.

【발명의 구성 및 작용】

<29> 이러한 목적을 달성하기 위한 본 발명에 따른 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템은 각각의 시스템마다 프라이머리부와 백업부를 두고; 이벤트 처리는 모든 시스템의 프라이머리에서 기본적으로 수행하면서 시스템이 비정상적인 상태를 대비하여 현재 시스템이 아닌 다른 시스템에 현재 프라이머리부에서 처리하는 이벤트 처리 데이터를 복사(백업)시키고; 시스템이 다운되는 경우가 발생되면 미리 백업 시켜놓은 시스템의 백업부를 프라이머리부로 변경하여 변경된 프라이머리부에서 이전의 프라이머리부에서 수행하던 이벤트를 계속 진행하고, 필요시 점유한 자원을 복구시키며 처리해야 할 이벤트가 더 이상 존재하지 않게 되면 변경된 프라이머리부의 동작을 중지하도록 이루어지는 특징이 있다.

<30> 이하, 본 발명이 속하는 분야에 통상의 지식을 지닌자가 본 발명을 용이하게 실시할 수 있도록 본 발명의 바람직한 실시예를 첨부된 도면을 참조하여 상세히 설명한다.

- <31> 도 2a는 본 발명의 일실시예에 따른 부하분담 구조와 프라이머리/백업 구조가 결합된 형태를 간략히 도시한 개념도이고, 도 2b는 도 2a와 같은 형태로 이루어진 시스템에서 이벤트 발생시 부하분담 및 백업 설정 관계를 도시한 블록도이다.
- <32> 도 2a를 참조하면, 본 발명의 일실시예에 따른 부하분담 구조와 프라이머리/백업 구조가 혼합된 시스템은 각 시스템마다 실장되는 프라이머리 1, 프라이머리 2, 프라이머리 3(110,210,310); 각 프라이머리(110,210,310)에 대응하여 구성되는 백업 1, 백업 2, 백업 3(120,220,320); 분산제어 환경(400); 구성요소 관리부(Configuration Management : 이하 'CM'이라 함)(500); 분산 알고리즘(600); 공유자원(이하 'Logical Shared Resource'라 함)(700); 및 이벤트(800)를 포함한다.
- <33> 상기와 같이 구성된 상태에서 각각의 시스템 A, 시스템B, 시스템C(100,200,300)는 CM(500)에 의해 각각 프라이머리 1, 프라이머리 2, 프라이머리 3(110,210,310)을 생성하고, 프라이머리 1, 프라이머리 2, 프라이머리 3(110,210,310)에 대해 각각 백업 1, 백업 2, 백업 3(120,220,320)을 구성한다.
- <34> 만약 새로운 시스템을 추가하는 경우, CM(500)에 의해 프라이머리 4(미도시)를 새로이 생성하고, 새롭게 생성된 프라이머리 4(미도시)에 대해 백업을 재구성하게 된다.
- <35> 이때 CM(500)은 프라이머리 간의 부하 분담 처리를 위한 인덱스(index)와 프라이머리/백업간의 구성을 위한 인덱스를 별도로 관리한다.
- <36> CM(500)은 프라이머리와 백업간의 구성 요소만을 관리하며, 이벤트 발생에 대해서는 관여하지 않는다.

- <37> 이벤트 발생 처리는 Logical Shared Resource(700)에서 담당하는데, 프라이머리 1, 프라이머리 2, 프라이머리 3(110,210,310)에 의해 공유되어 새로운 이벤트 발생시 프라이머리 1, 프라이머리 2, 프라이머리 3(110,210,310)에 의해 자원이 점유된다.
- <38> 상기와 같이 프라이머리/백업 구조가 결정이 되고, Logical Shared Resource(700)가 공유된 상태에서 새로운 이벤트가 발생하면, 프라이머리 시스템에서 처리하는 과정과 프라이머리 시스템이 다운되어 백업 시스템에서 이를 계속해서 처리하는 방식 및 자원을 복구하는 과정을 수행한다.
- <39> 먼저 프라이머리 지정 방식은 프라이머리 1, 프라이머리 2, 프라이머리 3(110,210,310) 순서대로 한번씩 이벤트를 분산하여 지정하는 라운드 로빈 방식(round robin)과, 시스템에 걸리는 부하를 고려하여 부하가 적은 쪽으로 분배하는 방식 등의 분산 알고리즘에 의해 결정된다.
- <40> 본 발명에서는 분산 알고리즘(600)에 따라 분배하는 방식을 예로 들어 설명하기로 한다.
- <41> 도 2b에 도시된 바와 같이, 분산 알고리즘(600)에 의해 해당 이벤트 처리를 위해 프라이머리 1(110)이 결정된 경우를 가정하면, 프라이머리 1(110)에서는 이벤트(800)를 수신하여 필요한 Logical Shared Resource(700)를 점유하고 이벤트처리를 진행한다.
- <42> 그리고 프라이머리 1(110)에서는 시스템 B(200)에 있는 백업 1(220)에 통보하여 프라이머리 1(110)과 백업 1(220)이 일치된 데이터를 유지하도록 한다.

- <43> 프라이머리 1(110)에서 백업 1(220)로 복사하는 데이터는 성능을 고려하여 복구 및 공유할 최소한의 데이터(예컨대, 이벤트에 대한 인덱스, 진행상태 및 점유자원)만 유지한다.
- <44> 새로운 이벤트가 발생하면 분산알고리즘에 의해 프라이머리 2, 프라이머리 3(210,310)으로 분배되고, 이에 대한 백업 2, 백업 3(320,120)은 백업 1(220)과 같이 이벤트는 처리를 하지 않고 복구에 필요한 데이터만 유지한다.
- <45> 이하 도 3a 및 도 3b를 참고로하여 특정 프라이머리가 다운된 경우를 설명한다.
- <46> 도 3a는 도 2b와 같은 관계에 따라 실제로 특정 프라이머리가 다운된 경우에 특정 백업부가 특정 프라이머리부로 변경되는 과정을 도시한 블록도이고, 도 3b는 도 2b와 같은 관계에 따라 실제로 특정 프라이머리가 다운된 경우에 남은 프라이머리의 부하분담 과정을 도시한 블록도이다.
- <47> 도 3a에 도시된 바와 같이, 프라이머리 1이 다운되는 경우 CM(500) 및 분산 제어 환경(400)에 의해 백업 1이 프라이머리 1'(220)로 변경된다. 이때 프라이머리 1'(220)은 새로운 이벤트를 처리하지 않고 프라이머리 1(110)에서 처리하던 이벤트를 계속 진행하고, 필요시 점유한 자원을 복구한다. 처리해야 할 이벤트가 더 이상 존재하지 않게 되면 프라이머리 1'(220)은 기능이 중지된다.
- <48> 새로운 이벤트가 발생하는 경우 도 3b에 도시된 바와 같이, CM(500) 및 분산 제어 환경(400)은 프라이머리 2, 3(210,310)에서만 처리하도록 하여 프라이머리 1(110)의 부하를 분담한다.

- <49> 한편 시스템 B(200)에서는 프라이머리 2(210)와 프라이머리 1'(220)가 존재하지만, 프라이머리 1'(220)는 프라이머리 1(110)이 처리하던 이벤트 및 복구에 대해 최소한의 기능만을 유지하므로 프라이머리 2,3(210,310)의 부하분담에 큰 영향을 미치지 않는다.
- <50> 이어 프라이머리 1(110)이 재동작하게 되면, 프라이머리 1'(220)는 다시 백업 1로 변경하고 프라이머리 1(110)은 부하분담에 합류하여 프라이머리 1,2,3(110,210,310)으로 부하가 균일하게 분배된다.
- <51> 상기의 과정에 따라 프라이머리 1,2,3(110,210,310)은 부하분담 처리를 수행함으로써 부하를 균일하게 분담하여 이벤트를 처리할 수 있어 자원의 낭비가 없고, 백업 1,2,3(220,320,120)은 프라이머리에 대해 완벽하게 이중화 구조를 제공하므로 시스템을 안정적으로 운영할 수 있게 된다.
- <52> 이상 본 발명의 바람직한 실시예에 대해 상세히 기술되었지만, 본 발명이 속하는 기술분야에 있어서 통상의 지식을 가진 사람이라면, 첨부된 청구 범위에 정의된 본 발명의 정신 및 범위를 벗어나지 않으면서 본 발명을 여러 가지로 변형 또는 변경하여 실시할 수 있음을 알 수 있을 것이다. 따라서 본 발명의 앞으로의 실시예들의 변경은 본 발명의 기술을 벗어날 수 없을 것이다.

【발명의 효과】

- <53> 이상 설명한 바와 같이, 본 발명은 프라이머리/백업 구조와 부하 분담 구조에서 개별적으로 제공하는 장점들만을 취하여 이루어진 구조로서, 다음과 같은 장점이 있다.
- <54> 첫째, 존재하는 모든 시스템을 사용하므로 자원의 낭비가 없다.

- <55> 둘째, 필요한 시스템을 쉽게 적용하므로 확장성이 좋다.
- <56> 셋째, 최소한의 시스템의 최대한의 성능을 발휘할 수 있으며 신뢰도를 보장하므로 경제성이 뛰어나다.
- <57> 넷째, 분산 시스템의 장점인 데이터의 공유 기능을 그대로 유지할 수 있다.
- <58> 다섯째, 1개의 시스템만으로도 프라이머리/백업 구조를 유지할 수 있다.
- <59> 상기의 장점을 요약하면 본 발명은 이벤트 발생에 대하여 부하 분담하여 처리하고 가용한 모든 시스템이 사용되므로 자원의 낭비없이 성능이 향상되는 구조이며, 백업 시스템으로 데이터가 복사되므로 시스템을 안정적이며 효율적으로 운영할 수 있는 효과가 있다.

【특허청구범위】**【청구항 1】**

물리적으로 분산된 다수의 시스템에 있어서,

이벤트들에 대해 부하분담 처리순서에 따라 부하를 분담하도록 각각의 시스템에
실장되는 프라이머리부;

프라이머리부가 다운되는 경우를 대비하여 복구에 필요한 최소한의 데이터만 프라
이머리로부터 수신하여 미리 지정된 시스템의 백업부에 동일한 데이터가 저장하도록 각
각의 시스템에 실장되는 백업부;

각각의 프라이머리부에 대한 백업부를 지정하고, 백업부에 대한 프라이머리부의
위치를 관리하는 구성요소 관리부;

이벤트 발생시 이벤트를 어느 시스템에서 처리할지를 결정하는 분산 알고리즘 처리
부;

각각의 시스템에서 공유해서 사용하고 프라이머리에서 점유하여 사용하는 자원들
중 반드시 복구되어야하는 공유자원;

분산 시스템에서 처리하고자 새로 생성되는 이벤트; 및

상기 프라이머리부 및 백업부로 이루어지는 각각의 시스템과 상기 구성요소 관리부
및 분산 알고리즘 처리부 그리고 공유자원간의 분산 처리가 가능하도록 하는 미들웨어
플랫폼으로 이루어지는 분산 제어 환경으로 이루어지는 것을 특징으로 하는 부하 분담
구조와 프라이머리/백업 구조가 혼합된 시스템.

【청구항 2】

제 1항에 있어서, 상기 각각의 프라이머리부에 대해 각각의 백업부가 할당되는 방식은

프라이머리부에 대한 백업을 위해 현재 시스템이 아닌 다른 시스템으로 백업부가 할당되는 것을 특징으로 하는 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템.

【청구항 3】

제 1항에 있어서, 상기 구성요소 관리부는

프라이머리부 간의 부하분담 처리를 위한 인덱스와 프라이머리부에 대해 지정된 백업부의 구성에 대한 인덱스를 관리하는 기능을 포함하여 이루어지는 것을 특징으로 하는 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템.

【청구항 4】

제 1항에 있어서, 상기 분산 알고리즘 처리부는

프라이머리 순서대로 이벤트를 분산하여 분배하는 라운드 로빈 방식으로 이루어지는 것을 특징으로 하는 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템.

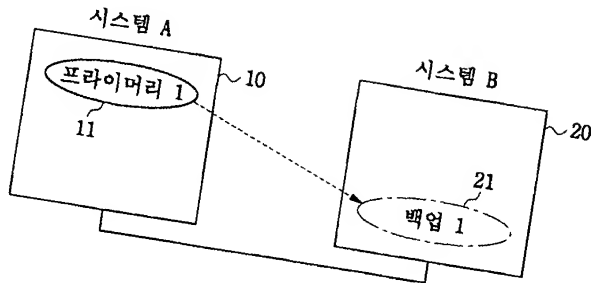
【청구항 5】

제 1항에 있어서, 상기 분산 알고리즘 처리부는

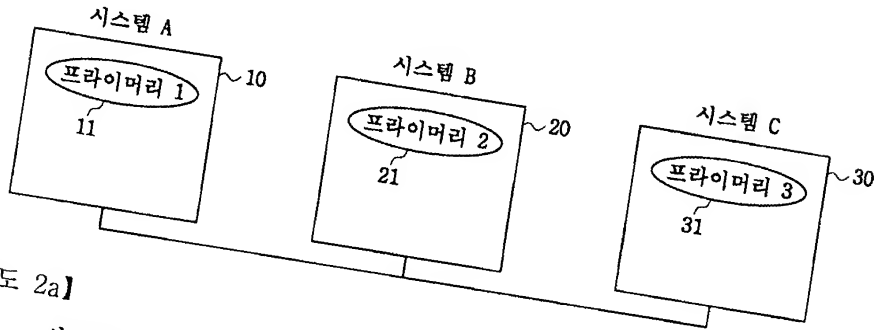
부하 분담을 계산하여 이벤트를 어느 시스템에서 처리할지를 결정하는 방식으로 이루어지는 것을 특징으로 하는 부하 분담 구조와 프라이머리/백업 구조가 혼합된 시스템.

【도 1a】

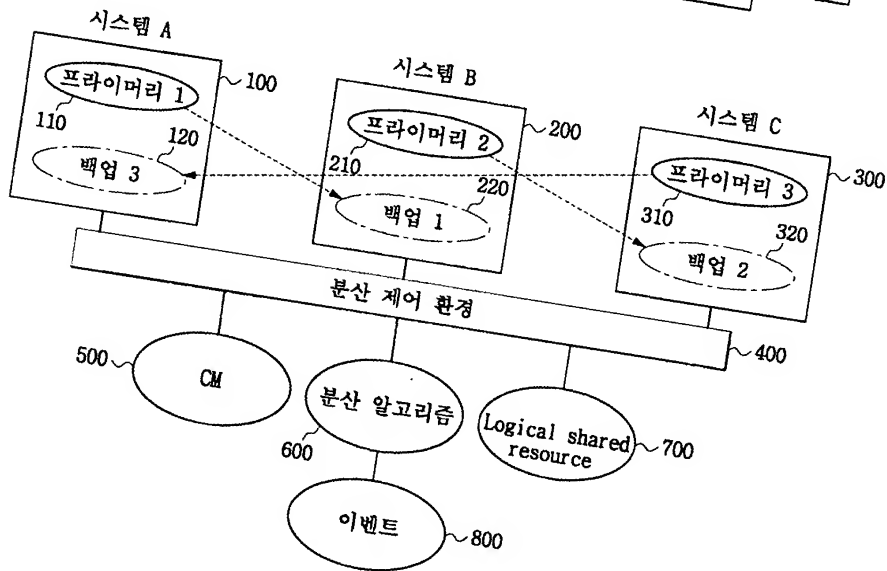
【도면】



【도 1b】

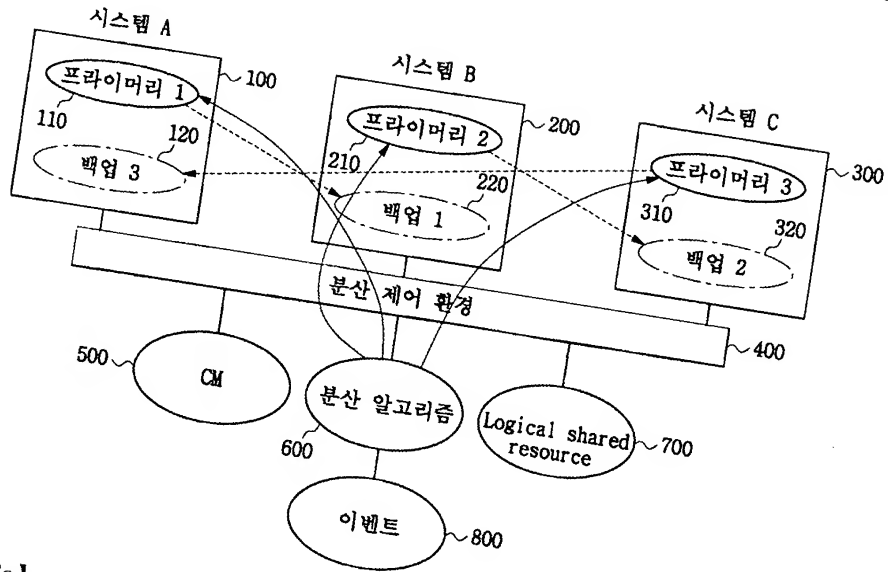


【도 2a】

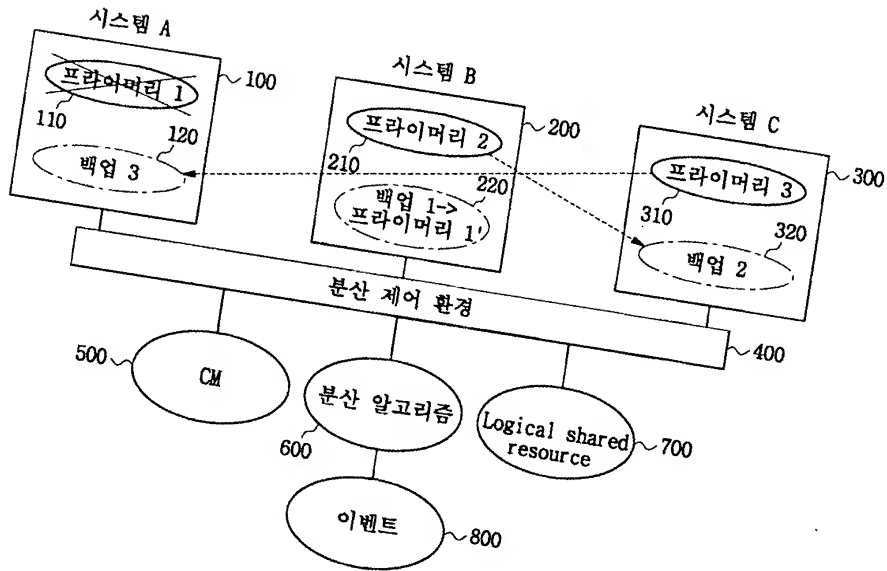


【도 2b】

출력 일자: 2003/3/3



【도 3a】



【도 3b】

출력 일자: 2003/3/3

